UNIVERSITÉ DE MONTRÉAL

MINIMAL FLEET SIZE IN TRANSHIPMENT-TYPE VEHICLE SCHEDULING PROBLEMS

by

David Scott*

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

MINIMAL FLEET SIZE IN TRANSHIPMENT-TYPE VEHICLE SCHEDULING PROBLEMS

by

David Scott*

* Department of Decision Sciences and Management Information Systems, Concordia University, Montréal (Québec) CANADA

Cette étude a été publiée grâce à une subvention du fonds F.C.A.R. pour l'aide et le soutien à la recherche.

Université de Montréal
Centre de recherche sur les transports - Publication #487
Mai 1986

Dépôt légal - 4e trimestre 1986

ABSTRACT

A number of authors have written on the problem of assigning a fixed schedule of trips to vehicles. Two objectives are commonly sought : that of minimum fleet size to serve all the trips, and that of minimum total operation cost. This problem can be modeled in a fairly standard way as a transhipment problem. This paper presents new results giving sufficient conditions that a solution to the more general problem of minimum operation cost will also be a solution which minimizes the fleet size.

- - - - - - - - - -

RÉSUMÉ

De nombreux chercheurs ont étudié le problème d'affectation aux autobus d'un nombre donné de voyages fixes. Dans ce problème, on cherche généralement à atteindre deux objectifs : le flot minimal de véhicules qui dessert tous les voyages, ainsi que le coût minimal d'exploitation. Ce problème se pose de façon standard, comme un problème de transbordement. Dans cet article, nous présentons les conditions suffisantes selon lesquelles une solution au problème de coût minimal d'exploitation est aussi une solution au problème de flot minimal.

Minimal fleet size in transhipment-type vehicle scheduling problems

David Scott

Department of Decision Sciences
and Management Information Systems
Concordia University
Montréal, Québec, Canada

Introduction

This paper discusses a somewhat familiar optimization problem, that of assigning vehicles to serve a fixed timetable of trips. A number of authors have written on this problem and have offered, on the one hand, algorithms which seek to minimize the total number of vehicles in a solution and, on the other hand, algorithms which seek to minimize the total capital and operating cost implied by a solution. Some authors (see, e.g., Bodin et al [4])have noted, however, that the twin objectives of minimum fleet size and minimum total capital and operation cost are not always compatible and that certain misspecification errors can occur when analysts try to achieve both simultaneously. This paper presents new results which show that there need not be any conflict between these two scheduling objectives, under circumstances which can generally be expected to apply in a large class of transportation scheduling problems. In particular, we give sufficient conditions that a solution to the problem of minimum operation cost will also be a solution which minimizes the fleet size.

The earliest attempts to solve the vehicle scheduling problem focussed on the minimum fleet size necessary to serve a fixed set of trips. Dantzig and Fulkerson [7] were the first to pose this problem as a linear program. Later, Dantzig and Hoffman [8] recognized a parallel structure between this scheduling problem and that of finding a minimal chain decomposition of a partially ordered set (Dilworth [11]). Ford and Fulkerson [12] drew a further parallel with the König-Egervary theorem in graph theory and proposed a simple solution algorithm. Raghavachari and Mote [16] extended this theory to the problem of minimal chain decomposition of a directed acyclic graph. Such a graph can be constructed without initially assuming a partially ordered set. These authors give public transport vehicle scheduling as an example of such a chain decomposition on an acyclic graph.

A somewhat different line of inquiry into the fleet size problem has examined the use of "deficit functions" (Gertsbach and Gurevich [14]; Ceder and Stern [6]; Stern and Ceder [19]). A deficit function is a step function defined at a terminal (a place where trips start or end). Each time a trip leaves a terminal, the deficit function increases by one at that terminal, and each time a trip arrives at a terminal the deficit function decreases by one. Following Bartlett [1] and Salzborn [17], Ceder and Stern [6,19] use deficit functions to obtain upper and lower bounds on the total fleet size required to serve a schedule and propose a heuristic for assigning vehicles to trips in such a way that the fleet size is minimized. Ceder and Stern [6] note that under general conditions these heuristics are equivalent to the algorithm of Ford and Fulkerson [12]. The deficit-function approach has the advantage, however, of being oriented toward graphical presentation, which provides the basis for straightforward manual modification of a vehicle solution. Such modifications would be necessary in case an automatic solution violates additional constraints such as those on manpower usage or the total length of a vehicle path, as polynomial-time exact solution techniques do not yet exist for these problems (see, e.g., Bodin, et al [4]).

Following in the tradition of linear-programming approaches to the vehicle scheduling problem, several authors (Bennington and Rebibo [2]; Orloff [15]; Gavish et al [13]) have presented models which minimize objectives more complex than just the fleet size. These models are motivated by the fact that in addition to fleet size, the time that vehicles spend away from the depot and the distance travelled during the course of a schedule also contribute to the cost of serving a set of trips. The linear programs presented by these authors each represent a problem of minimum-cost flow on an acyclic graph and allow nonnegative costs on all arcs in the graph. Like the chain decompositions described above, a solution to this problem of minimum-cost flow is a decomposition of the set of trips into chains of trips, each of which is to be served by a single vehicle. In order to accommodate fleet size into the model, however, a large cost is associated with each complete chain. This "fleet size penalty" is generally set several orders of magnitude larger than other arc costs, to reflect the proponderance of vehicle capital costs in the vehicle scheduling problem. The minimum cost flow problem set up in this manner will ensure that fleet size is minimized, as a large cost accompanies the entry of each vehicle into service. The secondary objective of minimum operation cost, however, is supposed to be met through the use of realistic arc costs.

Such linear-programming approaches to the scheduling problem have had a certain appeal

because of their conceptual simplicity and ease of implementation. In practical scheduling problems, however, a model which minimizes the number of vehicle chains may not serve the real goals of the optimization. This is because a single vehicle could conceivably cover two or more chains of trips. In the case of timetables in which the frequency of service varies with time of day, a vehicle solution in which vehicles are removed from service during periods of low trip frequency will normally reduce manpower costs and corresponds in general to the kind of solution favored by transport operators. Scott [18] has pointed out that a model which applies a fleet size penalty to each chain can induce a solution with artificially inflated vehicle time and distance, because the model includes a built-in bias against the creation of independent chains. Such a result is something of a distortion of the original scheduling objectives.

Desrosiers et al [9] have presented a scheduling model which takes into account the use of a single vehicle to serve many chains of trips. This model incorporates a structure which is quite a bit more complex than that of the other approaches we have cited. We show in this paper that under conditions which are characteristic of a broad class of vehicle scheduling problem there need not be any conflict, even in a simple scheduling model, between the two objectives of minimizing fleet size and minimizing the cost of operating a schedule. In other words there is never, in general, any need for the application of a fleet size penalty to the creation of independent chains when one intends to find a minimum-cost scheduling solution which uses a minimal fleet. We first restate, in a general form, the linear program which has served as model for a number of vehicle scheduling algorithms since Dantzig and Fulkerson [7]. We then demonstrate results which give the conditions under which an optimal solution to this linear program gives a solution which minimizes the fleet size.

The vehicle scheduling problem as a linear program

This section describes the linear program which has been applied to vehicle scheduling by various authors. The minimum fleet size algorithms of Dantzig and Hoffman [8] and Raghavachari and Mote [16] are special cases of this linear program, as are models for minimizing total operation cost proposed by such authors as Bennington and Rebibo [2], Orloff [15], Gavish et al [15], and Bodin et al [4].

Consider a timetable of t trips indexed by $i \in T = \{1, \ldots, t\}$ which are to be served by a number, to be determined, of identical vehicles. We assume that T is such that an ordering,

denoted by $\leq$, can be defined on this set. Following Raghavachari and Mote [16] we do not assume that $\mathbf{T}$ is partially ordered, although we do assume that $\leq$ is such that there are no ordered cycles. We obtain a strict ordering $<$ from $\leq$ by the convention that, for $i,j \in \mathbf{T}$, $i < j$ if $i \leq j$ and $i \neq j$. We let $\mathbf{A}$ denote the set of all (i,j) pairs such that $i \in \mathbf{T}$, $j \in \mathbf{T}$ and $i < j$.

The scheduling task is to effectuate a least-cost partition of $\mathbf{T}$ into ordered subsets or chains $C_1, C_2, \ldots, C_k$ each of which contains the indices of trips which can be performed in sequence by a single vehicle. Associated with each trip i is a fixed start time a_i and end time b_i. In addition, for the same vehicle to perform trip j after trip i, a conversion time r_{ij} must be allowed between the two trips to allow for mandatory waiting time or empty running between termini ("deadheading"). In other words, two trips i and j may feasibly be performed by the same vehicle only if

$$b_i + r_{ij} \leq a_j \qquad (1)$$

A number of authors (e.g., Ford and Fulkerson [12] and Raghavachari and Mote [16]) have let (1) define the ordering $<$ on $\mathbf{T}$, in the sense that $i < j$ if and only if (1) is true.

Several kinds of vehicle scheduling problems (public transport scheduling is an example) have the characteristic that a vehicle must perform an empty (deadhead) trip to and from a depot at the beginning and end of its service assignment. These empty trips would occur because the first in-service trip in the chain of such trips served by the vehicle does not begin at the depot, nor does the last trip in the chain end there. Also, the time to perform such a deadhead would naturally depend on the terminus at which a trip chain begins or ends. Thus we assume that if a vehicle comes "into service" to perform trip j, then there is a nonnegative entry time h_{*j} which must be allowed before this trip can be performed. Similarly, if a vehicle goes "out of service" immediately after performing trip i, then there is a nonnegative exit time h_{i*} which must be allowed. Put another way, we define times a_j' and b_i' such that a chain $C_r = \{i, \ldots, j\} \subset \mathbf{T}$, where $i < \ldots < j$, begins at time

$$a_j' = a_j - h_{*j} \qquad (2a)$$

and ends at time

$$b_j' = b_j + h_{j*}. \qquad (2b)$$

Since each trip is to be served once and only once by a vehicle we assume that the costs to actually carry out these trips are constant across partitions of $\mathbf{T}$. We assume, however, nonnegative costs associated with the passage of a vehicle from one trip to another, with entry of a vehicle into service, and with exit of a vehicle from service. For any admissible (i,j) pair, we define c_{ij} as the conversion cost associated with sequencing trip j to immediately follow trip i in a chain. Similarly, for all $j \in \mathbf{T}$, we define c_{*j} as the entry cost associated with bringing a vehicle into service before trip j and, for all $i \in \mathbf{T}$, define c_{i*} as the exit cost associated with taking a vehicle out of service after trip i. The scheduling problem becomes one of finding a sequencing of trips, using multiple vehicles, such that the total of all conversion, entry, and exit costs is a minimum.

We pose the scheduling problem formally as follows: Let

$$x_{ij} = \begin{cases} 1 & \text{if task } j \text{ immediately follows task } i \text{ in a chain;} \\ 0 & \text{otherwise.} \end{cases}$$

for all (i,j) pairs in the set $\mathbf{A}$. Also, for all $i \in \mathbf{T}$ and $j \in \mathbf{T}$, let

$$x_{*j} = \begin{cases} 1 & \text{if task } j \text{ is the first task in a chain,} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{i*} = \begin{cases} 1 & \text{if task } i \text{ is the last task in a chain} \\ 0 & \text{otherwise.} \end{cases}$$

The scheduling problem is then

$$\text{SP:} \quad \min \sum_{(i,j)\in\mathbf{A}} c_{ij}x_{ij} + \sum_{j\in\mathbf{T}} c_{*j}x_{*j} + \sum_{i\in\mathbf{T}} c_{i*}x_{i*} \qquad (3)$$

subject to

$$\sum_{i:(i,j)\in\mathbf{A}} x_{ij} + x_{*j} = 1 \quad \text{for all } j \in \mathbf{T} \qquad (4)$$

$$\sum_{j:(i,j)\in\mathbf{A}} x_{ij} + x_{i*} = 1 \quad \text{for all } i \in \mathbf{T} \qquad (5)$$

$$x_{ij} \geq 0 \quad \text{for all } (i,j) \in \mathbf{A}$$

$$x_{*j} \geq 0 \quad \text{for all } j \in \mathbf{T}$$

$$x_{i*} \geq 0 \quad \text{for all } i \in \mathbf{T}$$

Table 1 is a summary of published studies which have proposed the use of some variant of SP in vehicle scheduling. From each of these studies, we give the definition of an admissible (i,j) pair (the definition of the set $\mathbf{A}$) as well as the conversion costs c_{ij}, the entry costs c_{*j}, and the exit costs c_{*i}. That the maximization problems posed by Dantzig and Hoffman [8] and by Raghavachari and Mote [16] can be expressed as SP with all $c_{*j} = 1$, $c_{ij} = 0$ and $c_{i*} = 0$ is easily seen by solving for the x_{*j} in (4) and substituting these into (3).

INSERT TABLE 1 ABOUT HERE

The way in which the set $\mathbf{A}$ is constructed is clearly important since it determines the variable set in the linear program SP. As we now show, the use of SP as a scheduling model is inappropriate when <u>both</u> of the following conditions hold: when $\mathbf{A}$ is defined as the set of all (i,j) such that $i < j$ (according to the definition (1)), and when entry costs c_{*j} are set very high compared with the conversion costs c_{ij} in order to represent incremental vehicle usage.

Consider a simple bus transit trip timetable in Figure 1. This timetable contains trips between two terminals A and B. In addition, we give entry and exit times between the bus depot and the two terminals. We assume a "fleet size penalty" of 1000 which is applied to each c_{*j}. We also assume that any cost coefficients c_{ij} are computed as the time between the scheduled end of trip i and the scheduled beginning of trip j.

INSERT FIGURE 1 ABOUT HERE

Two solutions are provided in Figure 1, each of which involves the use of two vehicles. The first of these is a "natural" manual solution in which one of the vehicles is taken out of

service during a slack period when it is not needed. The second is the optimal solution to SP, even though the first uses 20 fewer vehicle minutes. The reason for this behavior is that the second uses only two chains of trips, while the first uses three. Since the presence of each chain induces an x_{*j} variable to unity, and since the c_{*j} associated with each of these includes the massive fleet size penalty, SP will naturally favor the two-chain solution.

Note as well that the second solution also corresponds to a "first-in-first-out" vehicle scheduling rule as well as to the "maximum slack tie breaking rule" in the deficit function heuristic of Ceder and Stern [6].

A model proposed by Desrosiers et al [9] does not present this problem, but uses a structure which is more complex than that of SP. We discuss this approach in a later section of this paper.

Visual inspection of Figure 1 might indicate that we can induce the "natural" solution to the scheduling problem by solving SP without the fleet size penalty. This is in fact the case, as we show in the next section.

Sufficient conditions for minimum vehicles in an optimal solution to SP

In this section we demonstrate results which give sufficient conditions under which an optimal solution to SP will also minimize the number of vehicles used in a solution. Such a vehicle could be used for one or more chains of trips; if more than one chain is covered by a vehicle, then this vehicle experiences an idle period between each pair of chains. We first introduce some new terminology. Two chains C_r and C_s are concurrent if any vehicle which covers C_r cannot also cover C_s. Similarly, two chains are nonconcurrent if a single vehicle can cover both these chains. Finally, the members of a set of three or more chains are mutually concurrent if any two chains in this set are concurrent. The problem of minimizing the fleet size is equivalent to that of finding a minimal set of mutually concurrent chains.

In addition, we define the composition of the set $\mathbf{A}$. For $i,j \in \mathbf{T}$, $(i,j) \in \mathbf{A}$ if and only if

$$b_j + r_{ij} \leq a_j \qquad (6a)$$

$$b_i' > a_j' \qquad (6b)$$

$$c_{ij} \leq c_{i*} + c_{*j} \qquad (6c)$$

Condition (6a) is equivalent to that expressed by (1). Condition (6b) requires, in effect, that two ordered trips must be close together in time. Condition (6c) excludes a potential link between two trips if it would always be less costly to return to the depot between such trips. In fact, the definition (6) immediately yields the following results:

Lemma 1: For $k,\ell \in \mathbf{T}$ such that $b_k \leq a_\ell$, there exists a solution to SP such k and ℓ are in nonconcurrent chains if and only if $b_k' \leq a_\ell'$.

Proof: ($\leftarrow$) Consider the feasible solution to SP in which $x_{*j} = 1$ for all j and $x_{i*} = 1$ for all i. In particular $x_{k*} = 1$ and $x_{*\ell} = 1$, so that there is in this solution a chain in which k is the last trip and another in which ℓ is the first trip. Since $b_k' \leq a_\ell'$ these two chains are nonconcurrent. ($\rightarrow$) Consider a solution to SP in which k and ℓ are in nonconcurrent chains. Since each of these chains has a first and last trip and since $b_k \leq a_\ell$, there must exist indices m and n such that $b_m' \leq a_n'$. It cannot be true that $b_m' < b_k'$, since if it were than we could route a vehicle from trip k to the depot via trip m and reduce the exit time from trip k. Thus $b_k' \leq b_m'$. Using a similar argument it can be shown that $a_n' \leq a_\ell'$. The desired result follows immediately. ∎

Lemma 2: For two trips $i,j \in \mathbf{T}$, if $(i,j) \in \mathbf{A}$ then in any solution to SP either i and j are in the same chain or are in concurrent chains.

Proof: If $(i,j) \in \mathbf{A}$ then $b_i' > a_j'$. From Lemma 1 there is no solution to SP such that i and j are in nonconcurrent chains. Thus either i and j are in the same chain or are in mutually concurrent chains. ∎

Figure 2 illustrates the three possibilities for any (i,j) pair in a solution to SP: either $(i,j) \in \mathbf{A}$, i and j are in the same chain, or i and j are in concurrent chains.

INSERT FIGURE 2 ABOUT HERE

We now prove the following theorem.

Theorem 1: Let the set $\mathbf{A}$ be composed of all pairs (i,j) such that the conditions (6) hold. Then an optimal solution to the linear program SP will include a minimal set of mutually concurrent chains.

Proof: By the Lemma, the problem of minimizing the number of concurrent chains is equivalent to maximizing the number of $(i,j) \in \mathbf{A}$ such that trips i and j are in the same chain. This maximization problem can be stated as follows:

$$\text{SP':} \quad \text{maximize} \quad \sum_{(i,j)\in \mathbf{A}} x_{ij} \tag{7}$$

subject to

$$\sum_{i:(i,j)\in \mathbf{A}} x_{ij} \leq 1 \quad \text{for all } i \in \mathbf{T} \tag{8}$$

$$\sum_{j:(i,j)\in \mathbf{A}} x_{ij} \leq 1 \quad \text{for all } j \in \mathbf{T} \tag{9}$$

$$x_{ij} \geq 0 \quad \text{for all } (i,j) \in \mathbf{A}$$

We show that an optimal solution to SP implies an optimal solution to SP'.

First note that (4) implies (8) and that (5) implies (9), since $x_{i*} \geq 0$ and $x_{*j} \geq 0$ for all $i,j \in \mathbf{T}$. It remains to be shown that (3) implies (7) under the conditions of the theorem.

A solution to SP in which $x_{*j} = 1$ for all $j \in \mathbf{T}$ and $x_{i*} = 1$ for all $i \in \mathbf{T}$ is clearly feasible. It is not optimal if $\mathbf{A}$ is non-empty, however, because by condition (6c) there exists an (i,j) pair such that $c_{ij} < c_{i*} + c_{*j}$. Hence setting $x_{ij} = 1$, $x_{i*} = 0$, and $x_{*j} = 0$ would also be feasible but would decrease the value of the SP objective function. We can then choose

another such pair in **A**, say (k,ℓ), where $x_{k*} = 1$ and $x_{*\ell} = 1$ and similarly reduce the value of the objective function by setting $x_{k\ell} = 1$, $x_{k*} = 0$, and $x_{*\ell} = 0$. We continue in this way until there are no further such pairs, at which point, because of constraints (8) and (9), no other x_{ij} can be set to 1 without setting some other to zero. Furthermore, condition (6c) implies that the objective function will never be decreased by reducing the number of (i,j) pairs such that $x_{ij} = 1$. Thus the optimal solution to SP will include a maximum number of (i,j) pairs such that $x_{ij} = 1$, and thus, equivalently, will include a minimal set of mutually concurrent chains. ∎

It is of interest to know whether the definition (6), by restricting the variable set **A**, induces an increase in the optimal value of SP. This is not the the case, as we now show.

Theorem 2: The optimal value of SP using a variable set defined using (6) is the same as that using a variable set defined using (1).

Proof: Since (6a) is equivalent to (1), it suffices to show that the application of (6b) and (6c) do not affect the optimal value of SP. We consider (6c) first. From the definition of SP, any (i,j) pair such that (6c) does not hold will never be in an optimal solution. Thus the application of (6c) will not affect an optimal solution. Now consider a pair (k,ℓ) which satisfy (6a) and (6c). Trip k will never be the last trip in a chain at the same time that trip ℓ is the first trip in a chain in any optimal solution, since if this were the case we could put $x_{k\ell}$ to one, x_{k*} and $x_{*\ell}$ to zero in SP and reduce the value of the objective function. Thus the further requirement (6b) that $b_{k'} > a_{\ell'}$, since it only concerns simultaneous exit from k and entry to ℓ (which is nonoptimal), will not affect the value of the optimal value of SP. ∎

The three conditions (6) on admissibility to the set **A** are in fact not stringent and correspond to circumstances commonly found in vehicle scheduling problems. (6a) follows from simple logistical considerations. (6c) excludes uneconomical links between trips far apart in time. (6b) will always follow from (6c) in cases where conversion costs are monotonically related to conversion time. The implications of these results are that under conditions generally found in vehicle scheduling problems there need not be any incompatibility between the objectives of minimizing total operation cost and that of minimizing the fleet size in

a simple scheduling model such as SP.

Computational considerations

As the introduction to this paper pointed out, the optimization problem SP has been studied in one form or another by a large number of authors for over thirty years. It seems fairly safe to say that the computation of a solution to SP depends on techniques which are very well understood. A natural solution method is as follows. Consider a graph $\mathbf{G} = (\mathbf{N},\mathbf{A}^*)$ where $\mathbf{N}$ is a set of nodes and $\mathbf{A}^*$ a set of directed arcs. Each trip $i \in \mathbf{T}$ is represented by two nodes in this set, say ξ_i and η_i. In addition, we add to $\mathbf{N}$ a node γ which represents the depot. Thus the set $\mathbf{N}$ contains $2t+1$ nodes.

We create the set $\mathbf{A}^*$ of arcs by defining an arc from γ to each ξ_i and from each η_i to γ, with arc costs c_{i*} and c_{*j} respectively. In addition, for each $(i,j) \in \mathbf{A}$ we add an arc from η_i to ξ_j with arc cost c_{ij}. Thus there are $2t + \text{card}(\mathbf{A})$ arcs in $\mathbf{A}^*$. We apply arc capacities of one on each of these arcs.

Finally, we add exogenous demands of one at each η_i and exogenous supplies of one at each ξ_j. At γ there is neither supply nor demand. The optimization problem is then one of transhipment, to satisfy exogenous supplies and demands at least cost.

Figure 3 is a diagram of SP set up as a transhipment problem. Such a transhipment problem can be solved using widely-available solution methods, such as the primal simplex method applied to network optimization problems (see, e.g., Bradley et al [3]).

INSERT FIGURE 3 ABOUT HERE

The total number of arcs in this problem depends on the composition of $\mathbf{A}$ and thus is data-dependent. However, the conditions (6) on admissibility to $\mathbf{A}$ serve to limit the size of this set and hence the size of the optimization problem SP. Combining (6a) and (6b) and using definitions (2) we have that for any $(i,j) \in \mathbf{A}$,

$$r_{ij} \leq a_j - b_i < h_{*j} + h_{i*}$$

where h_{*j} and h_{i*} have been defined above as the entry time to trip j and the exit time from trip i. Thus, for any $j \in \mathbf{T}$, there are a limited number of i such that (i,j) will be admissible to $\mathbf{A}$. We can therefore expect that the number of arcs will grow less than quadratically and in most cases linearly in t. This solves the problem expressed by Bodin, et al [4] who claimed that a complete specification of the vehicle scheduling problem would require $o(t^2)$ arcs.

Discussion

This paper has presented conditions under which a solution to a vehicle scheduling problem over a fixed schedule of trips, using a conventional model of minimum cost flow, will use a minimum number of vehicles. These conditions are likely to be present in many naturally-occurring scheduling situations where vehicle depots do not occur at the same place as the demand for vehicle service and hence vehicles must perform out-of-service movements to and from a depot, and where demand for service is uneven throughout the scheduling period. Public transport vehicle scheduling is a clear example of this kind of scheduling. Another example is the creation of vehicle schedules from partial bus routes in school bus scheduling (Bodin et al [4]). In principle, any scheduling problem based on the decomposition of an acyclic graph could possibly benefit from the simplifications discussed in this paper.

The "fleet size penalty" representing vehicle capital costs may therefore, under very general conditions, be dropped from a transhipment-type scheduling model without affecting the determination of the fleet size. Certain serious specification errors which may arise from the use of this penalty can thus easily be avoided.

Some authors have proposed alternate network-flow formulations which take explicit account of the possibility of multiple entries of a vehicle into service during the service period. The first of these, presented by Desrosiers et al [9] after Bokinge and Hasselstrom [5], is not reducible to and is somewhat more complex than SP. This model includes 2 depot nodes for each trip and series of directed arcs between depot nodes. There are thus $4t$ nodes and $4t + \text{card}(\mathbf{A})$ arcs in this formulation, as opposed to $2t$ nodes and $2t + \text{card}(\mathbf{A})$ arcs in SP.

Bodin et al [4] have noted that multiple entries and exits can be handled explicitly by

including elements in **A** correspond to successive trips by the same vehicle with a visit to the depot in between. While this is in fact a correct specification of the scheduling problem, the size of **A** will be on the order of the square of the number of trips.

The results presented here have assumed an acyclic structure among vehicle trips. Trip schedules in which the start and end times of all trips are known in advance present such an acyclic structure. In the case of nondegenerate "time windows" during which trips may begin this requirement is no longer automatically satisfied. More sophisticated combinatorial techniques, such as those proposed by Desrosiers et al [10], are required to handle such variable trip schedules.

Acknowledgements

This research has been funded in part by Operating Grant A6795 from the National Sciences and Engineering Research Council of Canada. I am grateful to Dr. Themistocles Politof and to Dr. François Soumis for many helpful suggestions.

References

1. Bartlett, T. An algorithm for the minimum number of transport units to maintain a fixed schedule. Naval Research Logistics Quarterly, 1957, 4, 139.

2. Bennington, G. E. and K. K. Rebibo. Overview of the RUCUS vehicle scheduling program (BLOCKS). In D. R. Bergmann and L. D. Bodin (eds.). Preprints of the workshop on automated techniques for scheduling of vehicle operators for urban public transportation systems. Washington, D. C.: U. S. Department of Transportation, 1975.

3. Bradley, G., G. Brown, and G. Graves. Design and implementation of large scale primal transhipment algorithms. Management Science, 1977, 24, 1.

4. Bodin, L., B. Golden, A. Assad, and M. Ball. Routing and scheduling of vehicles and crews: the state of the art. Computers and Operations Research, 1983, 10, 63.

5. Bokinge, U. and D. Hasselstrom. Improved vehicle scheduling through systematic changes in the time-table. European Journal of Operational Research, 1980, 5, 388.

6. Ceder, A. and H. Stern. Deficit function bus scheduling with deadheading trip insertions for fleet size reduction. Transportation Science, 1981, 15, 338.

7. Dantzig, G. and D. Fulkerson. Minimizing the number of tankers to meet a fixed schedule. Naval Research Logistics Quarterly, 1954, 1, 217.

8. Dantzig, G. and A. J. Hoffman. Dilworth's theorem on partially ordered sets. In H. W. Kuhn and A. W. Tucker (eds.). Linear Inequalities and Related Systems. Princeton, N.J.: Princetion University Press, 1956.

9. Desrosiers, J., F. Soumis, and M. Desrochers. Routes sur un réseau espace-temps. Proceedings of the meeting of the Administrative Sciences Association of Canada, Operations Research section, 1982, 3, 28. Also Publication no. 236, Centre de recherche sur les transports, Université de Montréal, 1982 (in French).

10. Desrosiers, J., F. Soumis, and M. Desrochers. Routing with time windows by column generation. Networks, 1984, 14, 545.

11. Dilworth, R. A decomposition theorem for partially ordered sets. Annals of Mathematics, 1950, 51, 161.

12. Ford, D. R. and D. Fulkerson. Flows in Networks. Princeton, N. J.: Princeton University Press, 1962.

13. Gavish, B., P. Schweitzer, and E. Shlifer. Assigning buses to schedulies in a metropolitan area. Computers and Operations Research, 1978, 5, 129.

14. Gertsbach, I. and Y. Gurevich. Constructing an optimal fleet for a transportation schedule. Transportation Science, 1977, 11, 20.

15. Orloff, C. S. Route constrained fleet scheduling. Transportation Science, 1976, 10, 149.

16 Raghavachari, M. and V. L. Mote. Generalization of Dilworth's theorem on minimal chain decomposition. Management Science, 1970, 16, 508.

17. Salzborn, F. Optimum bus scheduling. Transportation Science, 1972, 6, 137.

18. Scott, D. A method for scheduling urban transit vehicles which takes account of operation labor cost. Publication no. 365, Centre de recherche sur les transports, Université de Montréal, 1984.

19. Stern, H. and A. Ceder. An improved lower bound to the minimum fleet size problem. Transportation Science, 1983, 17, 471.

Table 1

Summary of parameters for SP reported by various authors

Author(s)	admissible (i,j)	c_{ij}	c_{*j}	c_{i*}
Dantzig and Fulkerson [7] [1]	$0 \leq a_j - b_i - r_{ij}$	0	1 [2]	0
Raghavachari and Mote [16] [1]	$0 \leq a_j - b_i$ [3]	0	1 [2]	0
Bennington and Rebibo [2]	$0 \leq a_j - b_i - r_{ij}$	linear function of conversion and idle times [4]	capital cost [5] plus linear function of entry time	linear function tion of exit time
Orloff [15]	$0 \leq a_j - b_i - r_{ij}$	linear function of conversion and idle times	capital cost [5] plus linear function of entry time	linear function of exit time
Gavish et al [13]	$k_1 \leq a_j - b_i - r_{ij} \leq k_2$ (k_1, k_2 constant)	linear function of conversion and idle times, plus penalties [6]	capital cost [5]	0
Bodin et al [4]	$0 \leq a_j - b_i - r_{ij}$	linear function of conversion and idle times and deadhead distance	capital cost [5] plus linear function of entry time and deadhead distance	linear function of exit time and deadhead distance

Notes:

1. SP using these costs is equivalent to Dilworth chain decomposition (Dantzig and Hoffman, 1956; Ford and Fulkerson, 1962); see Appendix.

2. Although entry costs exist, there is no account taken of entry or exit times.

3. End terminus for trip i and start terminus for trip j are assumed to be identical.

4. For idle time in excess of a fixed constant a heavy penalty is used.

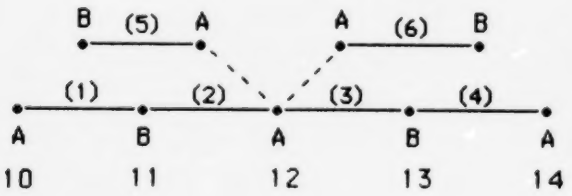
5. "Capital cost" is the incremental cost of an additional vehicle.

6. Penalties are added on links which do not already exist between trips and on links between trips from two different depots.

Figure 1

A sample trip timetable

#	Leave A	Arrive B	#	Leave B	Arrive A
(1)	10:00	11:00	(5)	10:30	11:30
(3)	12:00	13:00	(2)	11:00	12:00
(6)	12:30	13:30	(4)	13:00	14:00



Composition of **A**

(i,j)	c_{ij}
(1,2)	0
(2,3)	0
(3,4)	0
(5,3)	30
(2,6)	30

$c_{i*} = 1020$ for all i

$c_{*j} = 20$ for all j

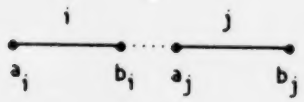
Two solutions to the scheduling problem

	sequence	time	SP objective value
Solution 1:	depot-(1)-(2)-(3)-(4)-depot	4:40	
	depot-(5)-depot	1:40	
	depot-(6)-depot	1:40	
	Total	8:00	3120
Solution 2:	depot-(1)-(2)-(6)-depot	4:10	
	depot-(5)-(3)-(4)-depot	4:10	
	Total	8:20	2140

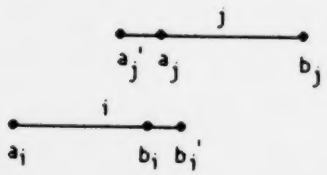
Figure 2

An illustration of chaining possibilities
for two trips i and j

$(i,j) \in A$:

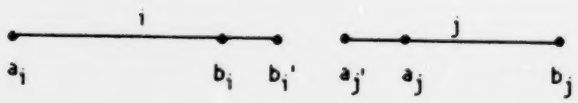


i,j in same chain



i,j in mutually concurrent chains

$(i,j) \notin A$:



i,j in nonconcurrent chains

Figure 3

Illustration of SP
transhipment graph